

Regression

732_Saksham

```
In [1]: import pandas as pd
import numpy as np
```

```
In [2]: df = pd.read_csv("Dataset/IceCreamData.csv")

df.head()
```

Out[2]:

	Temperature	Revenue
0	24.566884	534.799028
1	26.005191	625.190122
2	27.790554	660.632289
3	20.595335	487.706960
4	11.503498	316.240194

```
In [3]: df.corr()
```

Out[3]:

	Temperature	Revenue
Temperature	1.000000	0.989802
Revenue	0.989802	1.000000

```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 2 columns):
#   Column          Non-Null Count  Dtype  
---  -
0   Temperature     500 non-null   float64
1   Revenue         500 non-null   float64
dtypes: float64(2)
memory usage: 7.9 KB
```

```
In [5]: df.isnull().sum()
```

```
Out[5]: Temperature    0
Revenue              0
dtype: int64
```

```
In [6]: x = df['Temperature'].values.reshape(-1,1)

y = df['Revenue'].values.reshape(-1,1)

# because I want x and y values in 1-D array , not want Index col.
```

```
In [7]: x.shape
```

```
Out[7]: (500, 1)
```

```
In [8]: y.shape
```

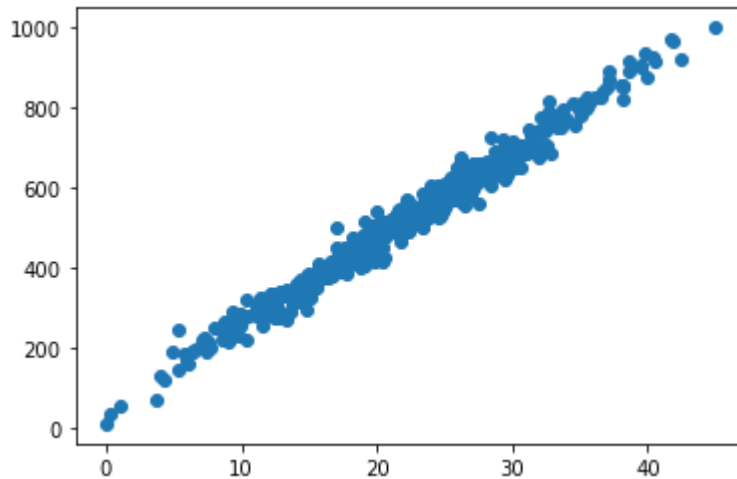
```
Out[8]: (500, 1)
```

```
In [9]: import matplotlib.pyplot as plt

import seaborn as sns
```

```
In [10]: plt.scatter(x,y)
```

```
Out[10]: <matplotlib.collections.PathCollection at 0x21fdbe24ac0>
```



```
In [11]: df.shape
```

```
Out[11]: (500, 2)
```

```
In [12]: from sklearn.model_selection import train_test_split
```

```
In [13]: x_train,x_test,y_train,y_test = train_test_split(x,y,random_state=0, test_size=0.1)
```

```
In [14]: x_train.shape
```

```
Out[14]: (450, 1)
```

```
In [15]: x_test.shape
```

```
Out[15]: (50, 1)
```

```
In [16]: # sns.regplot(x,y)
```

```
In [17]: from sklearn.linear_model import LinearRegression
```

```
In [18]: lr = LinearRegression()
```

```
In [19]: lr.fit(x_train,y_train)
```

```
Out[19]: LinearRegression()
```

```
In [20]: y_pred = lr.predict(x_test)
```

```
In [21]: prediction = pd.DataFrame(y_test, columns=['y_test'])  
prediction['x_test_col'] = x_test  
prediction['y_hat(predicted)'] = y_pred  
prediction['Residuals'] = y_test - y_pred
```

```
In [22]: prediction.head()
```

```
Out[22]:
```

	y_test	x_test_col	y_hat(predicted)	Residuals
0	704.281439	30.427792	697.707072	6.574367
1	632.901914	28.335363	652.739041	-19.837127
2	662.558990	28.865589	664.134040	-1.575050
3	449.813300	18.908489	450.147723	-0.334423
4	636.298374	28.900192	664.877682	-28.579308

```
In [23]: x_test
```

```
Out[23]: array([[30.42779184],
 [28.33536277],
 [28.86558895],
 [18.90848865],
 [28.90019172],
 [18.48314099],
 [25.11606991],
 [26.96421749],
 [28.99373705],
 [19.75470829],
 [23.38514451],
 [18.57811922],
 [26.9236056 ],
 [15.52116187],
 [15.02911176],
 [41.92444647],
 [39.5131548 ],
 [30.24724825],
 [23.34903419],
 [17.51707397],
 [16.14582413],
 [25.71796257],
 [11.13270573],
 [28.43656665],
 [15.67648661],
 [17.13279538],
 [15.21456942],
 [21.68442569],
 [20.26501213],
 [19.19495126],
 [27.72143999],
 [11.05909651],
 [12.57151377],
 [19.81463838],
 [23.98464085],
 [23.05621357],
 [12.27096675],
 [21.61064376],
 [24.5288527 ],
 [32.00436506],
 [18.43998163],
 [20.94791347],
 [24.34910395],
 [18.60275025],
 [40.47398918],
 [26.00519115],
 [23.15300185],
 [ 7.2613484 ],
 [30.22810362],
 [14.28719594]])
```

```
In [24]: m = lr.coef_
```

```
m
```

```
# slope
```

```
Out[24]: array([[21.49082669]])
```

```
In [25]: b = lr.intercept_
```

```
b
```

```
# y-intercept
```

```
Out[25]: array([43.78867085])
```

```
In [26]: # model equation :
```

```
# y = mx + b
```

```
# y = 21.4 x + 43
```

```
In [27]: y_1 = m * 30.42779184 + b
```

```
y_1
```

```
Out[27]: array([[697.70707182]])
```

```
In [28]: # prediction['x_test_col'] = x_test
```

```
# prediction.head()
```

```
In [29]: from sklearn.metrics import mean_squared_error
```

```
In [30]: mse = mean_squared_error(y_test, y_pred)
```

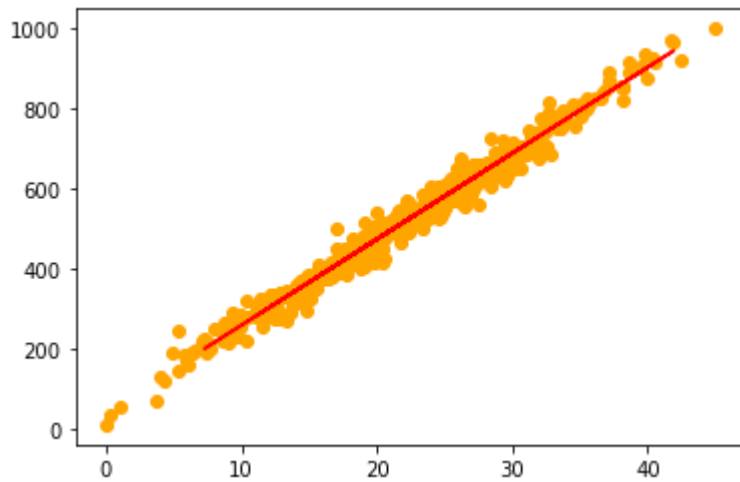
```
mse
```

```
# mean square error
```

```
Out[30]: 510.36278285590174
```

```
In [33]: plt.scatter(x,y, color = 'orange')  
  
plt.plot(x_test, y_pred, color = 'red')
```

Out[33]: [<matplotlib.lines.Line2D at 0x21fdcbab310>]



In []:

In []: